

You Just Might Be A Programmer!

by Robert Burns, Diablo Valley College

You don't need to be a math whiz...

You don't need to be Spock-like logic monster...

But...

- If you have a sense of "order", you just might be a programmer!
- If you are motivated by challenge, you just might be a programmer!
- If you like to solve puzzles, you just might be a programmer!
- If you are relentless in pursuing your goal, you just might be a programmer!
- If you have obsessive-compulsive disorder, you just might be a programmer!
- If you are an organization freak, you just might be a programmer!
- If you are a "detail person", you just might be a programmer!
- If you like machines more than you like people, you just might be a programmer!
- If you really don't mind that this web page is not pretty, you just might be a programmer!

Do you want to find out? Here's an exercise that will help you decide whether YOU might be a programmer: There will be three steps to this exercise:

1. Download and install a (free) C++ "compiler" over the Internet.
2. Write programs to simulate the roll of two six-sided **dice**, deal a **poker hand**, convert **Fahrenheit to Celsius**, and find the **area of a circle**.
3. Compile and run these programs.

This contains everything you need for writing computer programs in C++! It has Internet links for downloading and installing free software. It also has some programming exercises to get you started on your way to becoming a programmer.

First find a "text editor" on your computer -- you can use it to write your programs. Download and install the software you will need for "compiling" C++ programs, using the instructions provided below. Then type the computer language "code" for each of the programs listed below. Compile them into working "programs", and run them -- see what happens!

This may be enough to help you decide if programming is for you!

First Things First: Getting The "Compiler"

Here are the instructions for installing a compiler whether you use Linux, an Apple Mac, or a Windows PC:

For Linux Users

Linux users probably already have the GNU compiler installed. To find out if GNU is installed, go to a "command prompt" and enter the command `g++`. (That is, type `g++` and press the ENTER key.) If the command is recognized by Linux, as indicated by the message "no input files", then you have the GNU C++ compiler.



```
slack.dvc.edu - PuTTY
Linux 2.6.24.5-smp.
rburns@slack:~$ g++
g++: no input files
rburns@slack:~$
```

For Mac Users

First check to see if a compiler is already installed. Find the "Terminal" app -- it's in the Applications folder's Utilities folder. Start it up, and in the small, white window with the flashing "cursor", enter the command `g++`. If it says "command not found", then it's *not* installed, and you'll need to read the next paragraph. But if it says "no input files", then it's installed already!

To install, get XCode *for free* from the App store. To find it easily, go to <http://developer.apple.com/xcode>, and click the "View in Mac App Store" button. Then just download and install XCode as you would do for any other app. Once installed, you will *never have to run it* as you would run a normal app. But the first time you try to compile something using the instructions that follow, there should be a popup message guiding you to install "command line tools" and that will complete the installation process.

For Windows Users

First check to see if a compiler is already installed. Find the "command line" program in your start menu -- it's found in different places in different versions of Windows. Google "windows command prompt" if you need help finding it on your system. Start it, and in the small, black window with the flashing "cursor", enter the command `g++`. If it says "not recognized", then it's *not* installed, and you'll need to read the next paragraph. But if it says "no input files", then it's installed already!

Windows users can download the GNU C++ compiler *for free*! Just go to <http://tdm-gcc.tdragon.net/download> and use one of the links. There's one for newer, 64-bit systems -- it's probably the one you want. Once installed, there's really nothing to run as you would run a normal Windows program, like from an icon or menu. Just follow the instructions below, and they should work.

For Anybody, Anywhere, Anytime

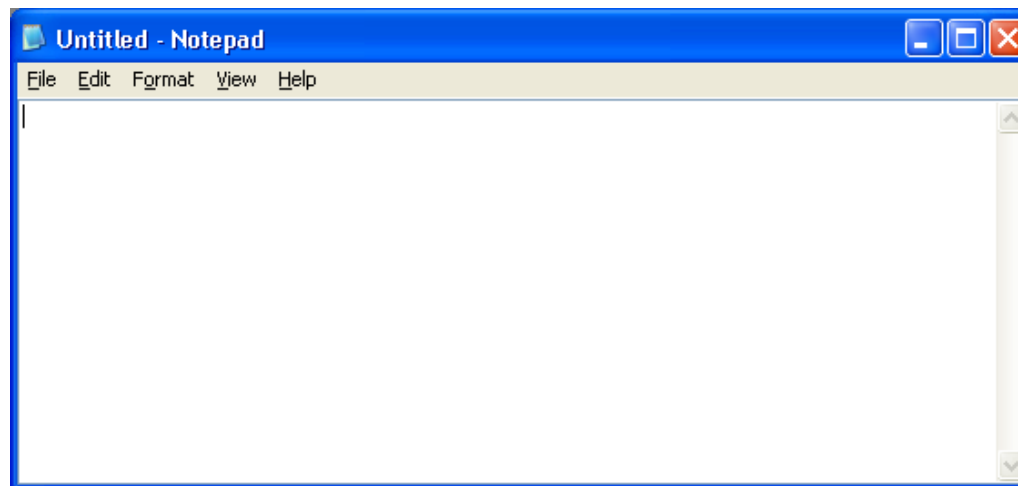
You can also write C++ programs using nothing but your Internet browser! Just go to www.compileonline.com/compile_cpp_online.php, where you'll see a place to write your code, a place to type your input (labeled "Command Line Arguments"), and button to run it (labeled "Compile & Execute"). If you're not quite sure about all this, then the online compiler is the way to go, because there's nothing to install.

Now you should be ready to write some programs!

First Program: Simulating The Roll Of Two Six-sided Dice

Okay -- we are ready to program! This program is one of the easy ones, because it requires no input from a user. You just run the program, and it simulates the roll of two dice. Run it again for another roll, etc.

You will need a text editor. Any text editor will do, including Windows **Notepad**, Linux **vi**, and Mac OSX **TextEdit**. These are *general-purpose text editors*, not specifically designed for use in programming, but they work just fine. (For details on setting up TextEdit on a Mac, go to http://support.apple.com/kb/TA20406?viewlocale=en_US.



(Word, Windows WordPad, and other editors capable of "rich text formatting" are not very suitable for writing code. They are page-oriented, and they embed formatting information into the files that they produce. While it is possible to configure such editors for text editing, it is better to avoid them.)

Type (or copy/paste) the following computer "code" into a text editor.

```
Rolling Two Six-sided Dice

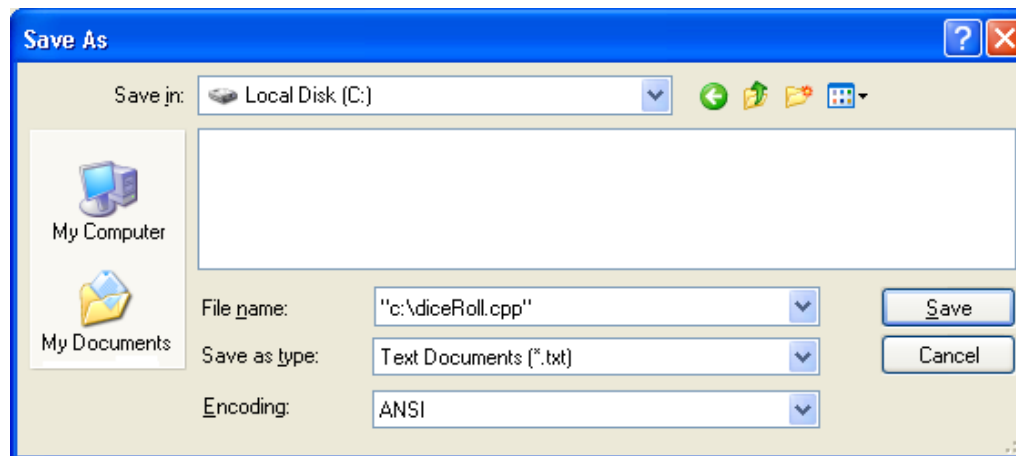
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main()
{
    srand(time(0)); rand(); // "seed" the random number generator

    int a; // the name of the first die
    int b; // the name of the second die
    int total; // the result
    a = 1 + rand() % 6;
    b = 1 + rand() % 6;
    total = a + b;

    cout << "Dice result: " << total
         << " (" << a << " and " << b << ")" << endl;
} // main
```

Save the file as **diceRoll.cpp**. In Windows, be sure to put **c:** in front of the file name, and enclose the whole thing in quote marks!



In Mac or Linux, save to your home folder.

NOTE: If you wish to save to a different folder, click [HERE](#) to watch a short video that explains how. Then proceed as normal.

Or if you are a Windows user and you have a flash drive, plug it in. Presuming it's the E drive, replace the `c:\` from "File name:" so that it says `"e:\diceRoll.cpp"`. When you get to the command line below, instead of the command `cd\`, type these two instructions instead: `e:` and then `cd\`. Then proceed as normal.

Now get a "command prompt" -- on a Mac, use the "Terminal" application. Get to it the same way you did when you checked to see if a compiler was already installed.

In the Windows command window, enter this command so that you can use the root folder of your c: drive (actually, you can use any drive and folder that you wish!):

```
cd\
```

Then enter these commands to compile and run your program -- and yes, uppercase or lowercase of letters *does matter!*:

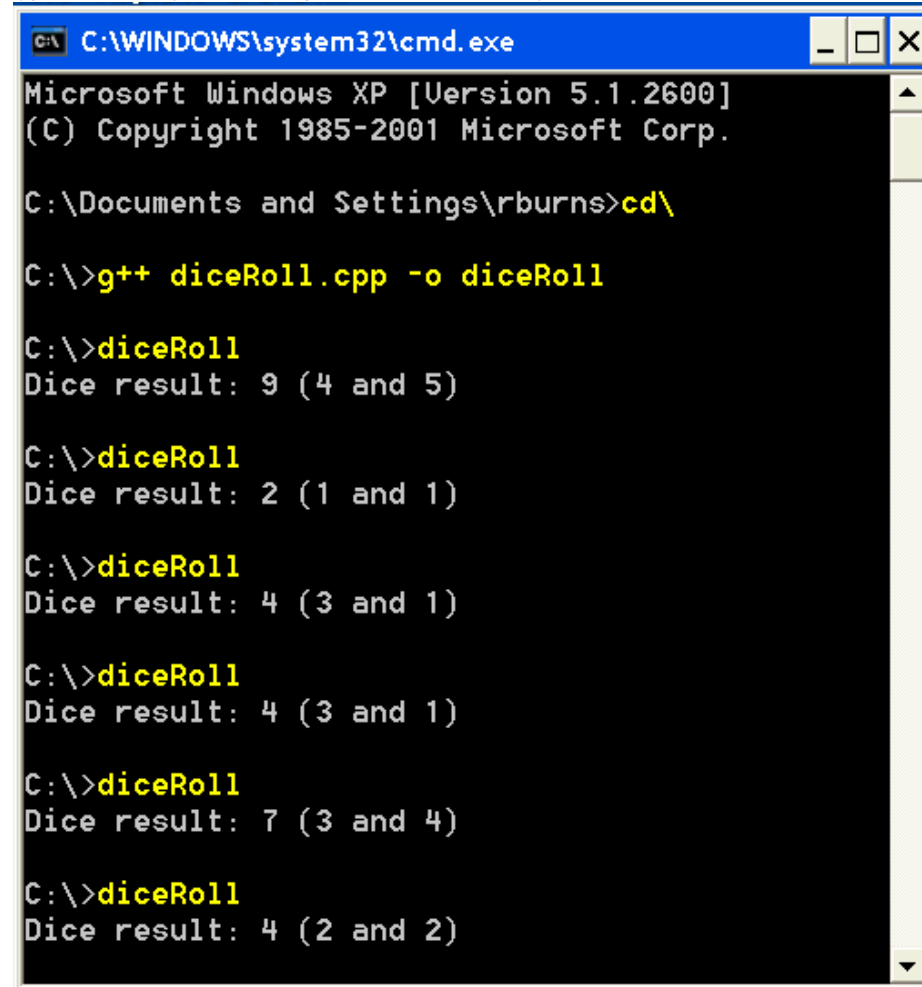
```
g++ diceRoll.cpp -o diceRoll
```

```
diceRoll
```

that's dash-OH, not dash-ZERO

Here's what it should look like for Windows (*be patient, Mac and Linux users -- your instructions are found just below this*):

(You type what's in yellow, and press the ENTER key to send the command to the computer.)



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\rburns>cd\

C:\>g++ diceRoll.cpp -o diceRoll

C:\>diceRoll
Dice result: 9 (4 and 5)

C:\>diceRoll
Dice result: 2 (1 and 1)

C:\>diceRoll
Dice result: 4 (3 and 1)

C:\>diceRoll
Dice result: 4 (3 and 1)

C:\>diceRoll
Dice result: 7 (3 and 4)

C:\>diceRoll
Dice result: 4 (2 and 2)
```

In Mac and Linux, the second command is different because of the added dot-slash. Remember to use it in each of the other examples below.

```
g++ diceRoll.cpp -o diceRoll
```

```
./diceRoll
```

that's dash-OH, not dash-ZERO

Did it work? That's your first C++ program!

Second Program: Converting Fahrenheit To Celsius

This might be interesting: if you can do this, then you can write programs to solve do lots of your physics and science problems. This program requires typed keyboard input from a user.

Use the instructions from the first program, and create the following file. Type (or copy/paste) the following computer "code" into a text editor.

```

Converting Temperature

#include <iostream>
using namespace std;

int main()
{
    double c; // degrees Celsius
    double f; // degrees Fahrenheit

    // ask user to enter Fahrenheit
    cout << "Enter the temperature in degrees Fahrenheit: ";
    cin >> f;

    c = 5 * (f - 32) / 9;
    cout << " That's " << c << " degrees Celsius!" << endl;
} // main

```

Save the file as **f2c.cpp**. In the Windows command window, enter these commands to compile and run your program:

```
g++ f2c.cpp -o f2c
f2c
```

On a Mac or in Linux, enter these commands to compile and run your program instead:

```
g++ f2c.cpp -o f2c
./f2c
```

Here's what it should look like for Windows:

(You type what's in yellow, and press the ENTER key to send the command to the computer.)

```

C:\WINDOWS\system32\cmd.exe

C:\>g++ f2c.cpp -o f2c

C:\>f2c
Enter the temperature in degrees Fahrenheit: 72
That's 22.2222 degrees Celsius!

C:\>f2c
Enter the temperature in degrees Fahrenheit: 32
That's 0 degrees Celsius!

C:\>f2c
Enter the temperature in degrees Fahrenheit: -40
That's -40 degrees Celsius!

C:\>_

```

Third Program: Calculating The Area Of A Circle

Ready for another? This program also requires typed keyboard input from a user.

Use the instructions from the first program, and create the following file.

Save the file as **area.cpp**. In the Windows command window,

Type (or copy/paste) the following computer "code" into a text editor.

```

Calculating Area

#include <iostream>
using namespace std;

#include <cmath>

int main()
{
    double r;
    double area;
    double pi = 4 * atan(1.0); // 4 times the angle whose tangent is 1

    // ask user to enter a circle's radius
    cout << "Enter the radius of a circle: ";
    cin >> r;

    area = pi * r * r;
    cout << " The area is " << area << endl;
} // main

```

enter these commands to compile and run your program:

```
g++ area.cpp -o area
area
```

On a Mac or in Linux, enter these commands to compile and run your program instead:

```
g++ area.cpp -o area
./area
```

Here's what it should look like for Windows:

(You type what's in yellow, and press the ENTER key to send the command to the computer.)

```

C:\WINDOWS\system32\cmd.exe

C:\>g++ area.cpp -o area

C:\>area
Enter the radius of a circle: 10
The area is 314.159

C:\>area
Enter the radius of a circle: 3
The area is 28.2743

C:\>_

```

Fourth Program: Dealing A Poker Hand

This program deals a hand of poker, with 5 cards. Like the first program, this one also requires no input from a user. You just run the program, and it lists 5 cards. Run it again for another deal, etc.

Use the instructions from the first program, and create the following file. Type (or copy/paste) the following computer "code" into a text editor.

Dealing A Poker Hand

Save the file as **pokerHand.cpp**. In the Windows command window, enter these commands to compile and run your program:

```
g++ pokerHand.cpp -o pokerHand
pokerHand
```

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main()
{
    srand(time(0)); rand(); // "seed" the random number generator

    int n = 5; // deal this many cards
    for (int i = 0; i < n; i++)
    {
        int value = 1 + rand() % 13;
        int suit = rand() % 4;

        switch (value)
        {
            case 1:
                cout << "Ace";
                break;
            case 11:
                cout << "Jack";
                break;
            case 12:
                cout << "Queen";
                break;
            case 13:
                cout << "King";
                break;
            default:
                cout << value;
        } // switch

        switch (suit)
        {
            case 0:
                cout << " of Spades" << endl;
                break;
            case 1:
                cout << " of Hearts" << endl;
                break;
            case 2:
                cout << " of Diamonds" << endl;
                break;
            case 3:
```

On a Mac or in Linux, enter these commands to compile and run your program instead:

```
g++ pokerHand.cpp -o pokerHand
./pokerHand
```

Here's what it should look like for Windows:

(You type what's in yellow, and press the ENTER key to send the command to the computer.)


```
    cout << " of Clubs" << endl;  
  } // switch  
} // for  
} // main
```

```
C:\WINDOWS\system32\cmd.exe  
  
C:\>g++ pokerHand.cpp -o pokerHand  
  
C:\>pokerHand  
9 of Clubs  
Jack of Clubs  
Ace of Diamonds  
3 of Clubs  
Jack of Spades  
  
C:\>pokerHand  
8 of Diamonds  
5 of Diamonds  
8 of Spades  
3 of Hearts  
10 of Hearts  
  
C:\>pokerHand  
3 of Diamonds  
3 of Clubs  
10 of Hearts  
Jack of Spades  
Queen of Hearts  
  
C:\>pokerHand  
King of Clubs  
Ace of Diamonds  
4 of Clubs  
King of Clubs  
10 of Hearts  
  
C:\>
```

That's it! That's the whole exercise. If you liked that, you just might be a programmer!

You may want to start your programming education at DVC. We have two possible starting places: COMSC-101 (formerly -100) or -110, depending on your aptitude and preparation going into this. If you did really well with the exercises presented here, and there's no holding you back, COMSC-110 may be your starting place. But if you had trouble getting off the ground with this, and you want to know more, COMSC-101 is definitely for you.

Here's a link to a diagram that shows the sequence of programming classes offered at DVC: www.dvc.edu/academics/mcsd/computer-science/pdfs/computer-science-sequence.pdf.